MVC

**MVC** é um jeito de organizar o código em três partes:

1. **Model (Modelo)**: Fica com os dados e a lógica de negócios. Exemplo: Salvar um cliente no banco de dados.
2. **View (Visão)**: É o que o usuário vê, tipo as telas ou páginas da aplicação. Exemplo: Formulário de cadastro de cliente.
3. **Controller (Controlador)**: Faz a ligação entre o que o usuário faz (clicando, preenchendo formulários) e o que deve acontecer no sistema (salvar, atualizar, etc). Exemplo: Quando o usuário clica em "salvar", o Controller manda os dados para o Model.

Resumindo:

* **Model** = Dados
* **View** = O que o usuário vê
* **Controller** = O que o usuário faz e o que o sistema responde.

A ideia é separar essas partes para o código ficar mais organizado e fácil de mexer!

# mvc no django

No **Django**, a arquitetura mais usada é chamada de **MVT (Model–View–Template)**, que é bem parecida com o clássico **MVC (Model–View–Controller)**, só muda um pouco a nomenclatura e a forma como o framework organiza as responsabilidades.

Um padrão de projeto bem conhecido é o MVC (Model, View, Controller) que é um [design pattern](https://www.devmedia.com.br/introducao-design-pattern/18838)(padrões de projetos) baseado em separação em três camadas os quais estão conectadas entre si e desempenham papéis para que sua aplicação funcione. Essa lógica é utilizada em linguagens bem famosas no mercado como Java, C#, Ruby e PHP.

A arquitetura MTV é uma derivação desde padrão de projeto. O que se altera é a nomenclatura de arquivos e quais as camadas interconectadas. Este modelo é utilizado dentro do framework Django, e a ideia deste artigo é mostrar como funciona dentro de um aplicação web montada dentro do framework de Python.

A definição de Model é a mesma para a arquitetura MVC e MTV, que é o arquivo que contém a estrutura lógica do projeto e funciona como um intermediário para manipular dados entre o banco de dados e a View. Dentro do arquivo Model é determinado quais tipos de dados, e como será armazenado dentro do seu banco e, como será exibido quando for requisitado pela View.

https://diandrasilva.medium.com/como-funciona-a-arquitetura-mtv-django-86af916f1f63

# 

# relação mvc mvt

**Model (MVC) = Model (Django)  
Controller (MVC) = View (Django)  
View (MVC) = Template (Django)**

https://dev.to/bocha/django-architecturemvc-vs-mvt-3c3d?utm\_source